**SUMMARY OSVVM**

**What is OSVVM?**

* **Verification Framework:** Provides test components, test scenarios, and transaction interfaces.
* **Verification Utility Library:** Includes constrained random testing, scoreboards, functional coverage, error tracking, and requirements tracking.
* **Verification Component Library:** Offers components for communication protocols such as AXI4, UART, and xMII.
* **Script Library:** Contains tool-independent scripts.
* **Co-Simulation:** Enables running software in a hardware simulator.
* **Test Reports:** Generates test outputs in HTML and JUnit XML formats.

**OSVVM Verification Framework**

* OSVVM Verification Framework included,Transaction Interface(record)-Transaction API(procedures), Verification Components and Test Sequencer.
* Transaction Interface, record that groups related data and control signals for communication in the testbench.
* Transaction API, set of procedures for sending and receiving transaction records between components.
* Verification Components implement interface signaling like communication.
* Test Sequencer call test cases.

**OSVVM’s Easy Approach**

* Different interfaces (AXI4, Avalon, UART, AxiStream, etc.) perform similar operations:
* Address-based interfaces use read/write
* Streaming interfaces use send/get
* Model Independent Transactions (MIT) in OSVVM provide a common structure for these operations
* MIT structure : Transaction Interface (record) - Transaction API (procedures)

Benefits of MIT:

* Speeds up Verification Component (VC) development
* Enables code reuse across similar VCs
* Simplifies test case creation
* Supports co-simulation with all MIT-based VCs
* Reduces and clarifies documentation effort